1. Explain CLR and its execution process
The Common Language Runtime (CLR) is a runtime environment that manages the execution of .NET code. It is responsible for executing code written in languages that target the .NET Framework, such as C# and Visual Basic.
The CLR provides a number of services to the code that it executes, including memory management, thread management, code execution, and exception handling. It also enforces type safety and security, ensuring that code cannot access memory or resources that it is not authorized to access.

2. What are C# variables and how to declare C# variables?
A variable is a named memory location that we can create in order to store data.
Declare: [data type] [variable name]
Initialization: variable name = value
String s1;
S1 = “Hello”
Combine declare and initialization
[data type] [vaiable name] = [value]
String s1 = “Hello”

3. What are value types? Give some examples of predefined value types in C#.
Simple, enum, struct, nullable, tuple.
There is a lot of simple types predefined in C# like char, int, long, float, double etc.

4. Give examples for reference types and what is the difference between value type and reference type.

String, dynamic, and object are reference types.
When we declare a value type, the data will be stored in the stack.
For reference type, the reference is stored in a stack and it will point the data to heap.

5. What is immutable? Is string immutable or mutable?
Immutable means cannot be changed. String is immutable. When modify the string, C# will create a new string.

6. What is the string intern pool in C# and why do we need the intern pool?
String intern pool is a table containing each unique string in your program. It can decrease the memory we use when creating duplicate string objects. Eventually it will help us to save memory

List some ways to Concatenate Strings In C#
String combines = s1+ s2;
String combines = string.Conca(s1,s2);

7. What is meant by casting a data type? What are the 2 kinds of data type conversions available in C#?

Casting a data type means converting a data type to another data type.
Implicity conversation & explicit conversation (data lost)

8. Will the following code compile? Why?
No. If casting from double to int, the precision will be lost so we need to do type casting - a bracket to make it.
Int myInt = (int)myDouble;

9. What is the dynamic type? How does it differ from var?
Dynamic type can escape the compile-time type checking, it resolves the type at run time. It changes the type based on the assigned value. The type danamic only exists at compile time, not run time.

Var variables are identified at compile time. Dynamic variables are identified at run time. Var variable must be initialized at the time of declaration. Declaration is not mandatory to get initialized. Var are statically type and cannot be assigned different type. Dynamic variables are dynamic typed.

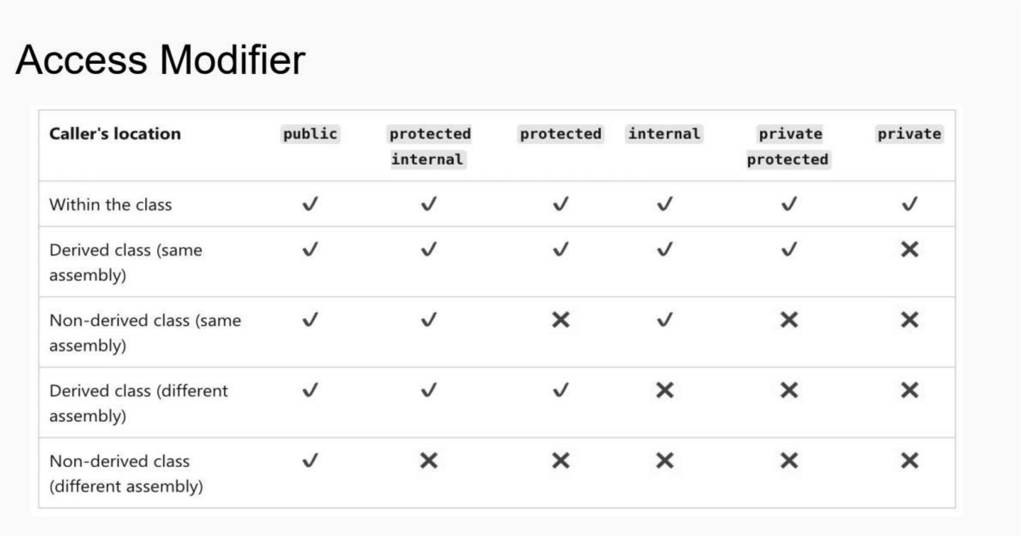
10. What is the nullable value type?
Extensions of all other value types with a null value.It gives another possible value to null or

11. What are the results of the following expressions? Please include the calculation process. (logical operator)

5&6 = 101 & 110 = 100 = 4
5|6 = 101 | 110 = 111 = 7
5^6 = 101 ^ 110 = 011 = 3

12. Why do we need to use the break statement in the Switch statement?
When we meet a unique case, we execute it and period. It will have encounter compile error.

13. What are access modifiers and their corresponding scopes in C#?
Access modifiers are keywords used to specify the declare field, method, constructor and class.



Access Modifier

Caller's location	`public`	`protected internal`	`protected`	`internal`	`private protected`	`private`
Within the class	✓	✓	✓	✓	✓	✓
Derived class (same assembly)	✓	✓	✓	✓	✓	✕
Non-derived class (same assembly)	✓	✓	✕	✓	✕	✕
Derived class (different assembly)	✓	✓	✓	✕	✕	✕
Non-derived class (different assembly)	✓	✕	✕	✕	✕	✕

14. Can you specify an access modifier for enumeration members?
We

15. What is assembly?
An assembly is a collection of types and resources that are built to work together and form a logical unit of functionality. Assembly is the scope of project.
.exe .dll are physical assemblies.